AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A multi-mode processor comprising:

a first instruction set engine to process instructions from a first instruction set architecture (ISA), the first ISA designed for a first processor having a first word size that defines the maximum number of bits that the first processor can handle as a single unit;

a second instruction set engine to process instructions from a second ISA, the second ISA designed for a second processor having a second word size that defines the maximum number of bits that the second processor can handle as a single unit, a bit length of the second word size being ~~different~~ greater than a bit length of the first word size;

a mode identifier;

a plurality of floating-point registers shared by the first instruction set engine and the second instruction set engine; and

a floating-point unit coupled to the floating-point registers, the floating-point unit including:

~~processing an input responsive to the mode identifier~~ pre-processing hardware to bypass an arithmetic unit if a token exists in the input and the mode identifier indicates the second ISA mode;

the arithmetic unit to process the input to produce an arithmetic result unless the input bypasses the arithmetic unit; and

post-processing hardware to perform a token specific operation if a token exists in the input and bypasses the arithmetic unit to produce an output.

2. (Previously Presented) The multi-mode processor of Claim 1 wherein the mode identifier is one of a plurality of bits in a processor status register.

3. (Cancelled)

4. (Previously Presented) The multi-mode processor of Claim 1 wherein the input includes data stored in at least one of the floating-point registers.

5. (Previously Presented) The multi-mode processor of Claim 1 wherein the input may contain a token, wherein the floating-point registers are 82 bits wide, and wherein the token being an 82 bit processor known value.

6. (Previously Presented) The multi-mode processor of Claim 3 wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

7. (Previously Presented) The multi-mode processor of Claim 1 wherein the floating point registers each comprise:

a sign bit,

an exponent; and

a significand.

8. (Previously Presented) The multi-mode processor of Claim 1 wherein the mode identifier indicates whether the processor is in a first mode or a second mode.

9. (Previously Presented) The multi-mode processor of Claim 1 wherein the mode identifier indicates whether the processor is in a 32 bit word ISA mode or a 64 bit word ISA mode.

10. (Currently Amended) A method in a multi-mode processor comprising:

fetching an input from at least one of a plurality of floating-point registers;

detecting whether the input contains a token;

if the token is detected in the input, checking what mode the processor is in;

when the mode the processor is in is a first word size instruction set architecture (ISA) mode, ~~processing~~ providing the input to an arithmetic unit ~~the input~~ to render an arithmetic result, the first ISA designed for a first processor having the first word size that defines the maximum number of bits that the first processor can handle as a single unit;

when the mode the processor is in is a second word size ISA mode, ~~performing~~ bypassing the arithmetic unit to post-processing hardware to perform a token specific operation, the second word size ISA designed for a second processor having the second word size that defines the maximum number of bits that the second processor can handle as

a single unit, a bit length of the second word size being ~~different~~ greater than a bit length of the first word size; and

producing an output based upon the mode the processor is in.

11. (Previously Presented) The method of Claim 10 wherein the input is comprised of at least one operand and at least one operator; wherein detecting comprises examining the at least one operand to determine whether any of the operands correspond to the token; and wherein checking comprises examining a mode identifier to determine whether the processor is in the first mode or the second mode.

12. (Previously Presented) The method of Claim 10 wherein processing comprises executing at least one operation on the at least one operand according to the at least one operator to achieve a result.

13. (Original) The method of Claim 10 wherein performing comprises propagating the token; and wherein producing output comprises setting the output to be the token.

14. (Original) The method of Claim 10 wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

15. (Original) The method of Claim 10 wherein checking comprises checking a mode identifier.

16. (Original) The method of Claim 10 wherein checking comprises checking a mode identifier bit in a processor status register.

17. (Original) The method of Claim 11 wherein the first mode is a 32 bit word ISA mode and the second mode is a 64 bit word ISA mode.

18. (Cancelled)

19. (Cancelled)